WHAT IS CLAIMED IS:

1. A method for transferring data in a data processing unit comprising:

receiving a first address;

if the first address contains a predetermined address component, then producing a second address and accessing a memory location at the second address; and

if the first address does not contain the predetermined address component, then accessing a memory location at the first address,

the first address comprising a plurality of address bits, wherein the predetermined address component comprises less than all of the address bits of the first address,

the first address and the second address having a common address component.

2. The method of claim 1 wherein the predetermined address component comprises the high order byte of the first address.

3. The method of claim 1 further comprising executing a virtual machine in a first page of an address space of the data processing unit, the first address being an address in the first page, wherein the second address is an address in another page of the address space of the data processing unit.

4 The method of claim 1 wherein an address space of the data processing unit comprises a plurality of pages, the first address being an address in a first page, the second address being in an address in a second page that is different from the first page.

5. A data processor comprising a memory management unit for handling addresses, the memory management unit comprising:

address input lines for presenting an input address;

address translation logic coupled to at least some of the address input lines; and

address output lines coupled to the address translation logic for presenting an output address,

the address translation logic operable to receive a first input address from the address input lines and to generate a first output address on the address output lines that is different from the first input address if a portion of the first input address contains a predetermined value.

6. The data processor of claim 5 wherein the address translation logic is further configured to detect a predetermined value present in the high order bits of the input address.

7. The data processor of claim 5 wherein the memory management unit further comprises a translation table, the translation table comprising M-bit values, the translation table addressed by an N-bit value, where M > N, wherein the translation table produces one of the M-bit values based on N lines of the address input lines, wherein the M-bit value produced is used in M lines of the address output lines.

8. The data processor of claim 7 wherein the remaining lines in the address output lines come from the address input lines.

9. A method for transferring data in a data processing unit comprising:

accessing a first value from a first data storage component of the data processing unit;

combining the first value with a base address to produce a first address, the base address representative of an area in a memory;

accessing a second value from a second data storage component of the data processing unit;

accessing a third value from a third data storage component of the data processing unit;

combining the second value with a value stored in a location in the memory indicated by the first address to produce a fourth value;

comparing the third value with the fourth value to produce a comparison result; and

based on the comparison result, either accessing the memory beginning at a location indicated by a second address or performing an exception operation, the second address being computed from the base address.

10. The method of claim 9 further including one or more steps of storing the first value in the first data storage component, storing the second value in the second data storage component, and storing the third value, wherein the one or more steps of storing are performed only at the time the data processing unit is initialized.

11. The method of claim 9 wherein one or more of the first value, the second value, and third value is stored respectively in the first data storage component, the second data storage component, and the third data storage component in a non-volatile manner.

12. The method of claim 9 further comprising loading one or more of the first data storage component, the second data storage component, and the third data storage component respectively with the first value, the second value, and third value.

13. The method of claim 9 wherein the second address is computed based on the base address and an offset value.

14. The method of claim 13 wherein the second address is computed by adding together the base address and the offset value.

15. The method of claim 9 wherein the exception operation comprises accessing content of a fourth data storage component, wherein subsequent processing by the data processing unit continues from instructions beginning at a location in memory computed based on the content of the fourth data storage component.

16. The method of claim 15 wherein the location in memory where subsequent processing continues is computed by adding the content of the fourth data storage component with an offset value.

17. The method of claim 9 wherein the area in memory represents a data object.

18. The method of claim 9 wherein the step of accessing is a step of copying the accessed content to another area in the memory.

19. The method of claim 9 wherein the step of accessing is a step of writing information to the accessed content.

20. The method of claim 9 wherein the first address is produced by adding together the first value and the base address.

21. A data processing unit comprising:

a memory;

instruction decoder logic for producing control signals in response to decoding program instructions;

data processing logic coupled to receive the control signals and configured to perform logic operations in accordance with the control signals;

an instruction register coupled to the memory and to the instruction decoder, the instruction register operable to store an instruction to be decoded;

a program counter operable to store an address of an instruction to be decoded;

a vector address data area for storing a vector address;

a first data area for storing a first value;

a second data area for storing a second value; and

a third data area for storing a third value,

the instruction decoder configured to assert first control signals in response to decoding a first instruction,

the data processing logic, in response to the first control signals, configured to:

combine the first value with a first operand contained in the first instruction to produce a first address;

combine the second value with a value stored in the memory at a location indicated by the first address to produce a fourth value;

compare the third value with the fourth value to produce a comparison result; and

based on the comparison result, either access the memory beginning at a location indicated by a second address or update the program counter with an address based on the vector address, the second address being computed from the first operand.

22. The data processing unit of claim 21 wherein the memory is a re-programmable, non-volatile memory.

23. The data processing unit of claim 21 wherein the memory is one of an EEPROM (electrically erasable programmable read-only memory) and flash memory.

24. The data processing unit of claim 21 wherein the data processing logic, in response to initialization of the data processing unit, is further configured to load data into one or more of the vector address data area, the first data area, the second data area, and the third data area.

25. The data processing unit of claim 21 wherein one or more of the vector address data area, the first data area, the second data area, and the third data area comprise non-volatile memory.

26. The data processing unit of claim 25 wherein the non-volatile memory is one of EEPROM (electrically erasable programmable read-only memory) or PROM (programmable read-only memory).

27. The data processing unit of claim 21 wherein one or more of the vector address data area, the first data area, the second data area, and the third data area comprise hard-wired logic.

28. The data processing unit of claim 21 wherein the instruction decoder is further configured to assert a second set of control signals in response to decoding a second instruction, the data processing logic, in response to the second control signals, is further configured to load data into one or more of the vector address data area, the first data area, the second data area, and the third data area, the data contained in the second instruction.

29. The data processing unit of claim 21 wherein the data processing logic, in response to the first control signals, is further configured to combine the first operand with a second operand contained in the first instruction to produce the second address.

30. The data processing unit of claim 29 wherein the first operand and the second operand are summed to produce the second address.

31. A memory access method in a data processing unit comprising re-programmable non-volatile memory, the method comprising:

detecting a write operation to an area in the re-programmable non-volatile memory;

determining a caching location which identifies an area of memory in a dynamic memory to which data to be written by the write operation can be cached, the area of memory identified by the caching location referred to as a caching area; and

writing the data to the caching area instead of writing the data to the area in the re-programmable non-volatile memory,

wherein the caching area is associated with the area in the re-programmable non-volatile memory.

32. The method of claim 31 wherein the step of determining a caching location comprises:

determining if the data can be stored in the dynamic memory, and if not then:

identifying one or more memory locations in the dynamic memory; and

writing content in the one or more memory locations to the re-programmable non-volatile memory; and

providing an address in the dynamic memory as the caching location in the area of memory.

33. The method of claim 32 wherein the step of determining a caching location in the dynamic memory is limited to determining a caching location in a segment of the dynamic memory.

34. The method of claim 31 further including detecting a second write operation to the area in the re-programmable non-volatile memory and writing the data associated with the second write operation to the area of memory identified by the caching location instead of writing the data to the area in the re-programmable non-volatile memory.

35. The method of claim 31 further including detecting a second write operation to a second area in the re-programmable non-volatile memory, determining if there is a caching area in the dynamic memory that is associated with the second area and if there is then writing data associated with the second write operation to the caching area associated with the second area.

36. The method of claim 31 wherein an address space of the re-programmable non-volatile memory is equal to or greater than an address space of the dynamic memory.

37. The method of claim 32 wherein the step of identifying one or more memory locations is based on the size of the data.

38. The method of claim 32 wherein the step of identifying one or more memory locations is based on contents of the data.

39. The method of claim 38 wherein the step of identifying one or more memory locations includes applying a hash function on the data to produce a hash result, the one or more locations determined based on the hash result.

40. The method of claim 31 wherein the re-programmable non-volatile memory is one of an EEPROM (electrically erasable programmable read-only memory) and a flash memory.

41. The method of claim 31 further including obtaining an address range which defines the area of memory in the dynamic memory.

42. The method of claim 31 further including detecting a read operation from the re-programmable non-volatile memory and reading the re-programmable non-volatile memory to effect the read operation.

43. The method of claim 31 further including detecting a read operation from the re-programmable non-volatile memory and reading one or more memory locations from the area of memory in the dynamic memory to effect the read operation.

44. A data processing unit comprising:

a bus for accessing a dynamic memory;

a re-programmable non-volatile memory;

memory access logic operatively coupled to the memory bus and to the re-programmable non-volatile memory; and

processor logic operatively coupled to the memory access logic to transfer data with the dynamic memory via the memory bus and to transfer data with the re-programmable non-volatile memory,

the memory access logic configured to detect a write operation to the re-programmable non-volatile memory,

the processor logic configured to respond to the memory access logic detecting the write operation and to:

identify a caching location in the area of memory wherein data to be written by the write operation can be cached; and

store the data in the area of memory identified by the caching location.

45. The data processing unit of claim 44 wherein the memory access logic is further configured to:

determine if the data can be stored in the area of memory; and

provide an address in the area of memory as the caching location in the area of memory,

wherein if it is determined that the data cannot be stored in the area of memory, then:

identify one or more memory locations in the area of memory; and

write content in the one or more memory locations to the re-programmable non-volatile memory,

wherein the one or more memory locations are available for caching a write operation.

46. The data processing unit of claim 44 wherein an address space of the re-programmable non-volatile memory is equal to or greater than an address space of the dynamic memory.

47. The data processing unit of claim 44 wherein the re-programmable non-volatile memory is one of an EEPROM (electrically erasable programmable read-only memory) and a flash memory.

48. The data processing unit of claim 44 wherein the dynamic memory is a random access memory.

49. The data processing unit of claim 44 wherein the memory access logic is further configured to detect a read operation from the re-programmable non-volatile memory and to effect a read operation therefrom.

50. The data processing unit of claim 44 wherein the memory access logic is further configured to detect a read operation from the re-programmable non-volatile memory and to access the area in memory in the dynamic memory to effect the read operation.

51. A memory access method comprising:

detecting a write operation to a re-programmable non-volatile memory; and

if a destination address associated with the write operation is within a first range of addresses, then performing a fast write operation of data associated with the write operation to the re-programmable non-volatile memory.

52. The method of claim 51 wherein the first range of addresses spans the entire address space of the re-programmable non-volatile memory.

53. The method of claim 51 wherein if the destination address is not within the first range of addresses, then performing a slow write operation of the data to the re-programmable non-volatile memory.

54. The method of claim 51 wherein the first range of addresses spans a range of addresses less than the address space of the re-programmable non-volatile memory.

55. The method of claim 51 wherein the step of performing a fast write operation is performed if the destination address falls within any of a plurality of ranges of addresses.

56. A memory access method comprising:

detecting a write operation to a non-volatile memory;

determining an access mode;

if the access mode is a first mode, then performing a fast write operation of data associated with the write operation to the non-volatile memory;

if the access mode is a second mode, then performing a slow write operation of the data associated with the write operation; and

if the access mode is a third mode, then:

if the destination address associated with the write operation is within a first range of addresses, then performing a fast write operation of the data associated with the write operation to the non-volatile memory;

if the destination address is not within the first range of addresses, then performing a slow write operation of the data to the non-volatile memory; and

if the access mode is a fourth mode, then:

determining a caching location which identifies an area of memory in a dynamic memory to which data to be written by the write operation can be cached, the area of memory identified by the caching location referred to as a caching area; and

writing the data to the caching area instead of writing the data to the area in the non-volatile memory,

wherein the caching area is associated with the area in the non-volatile memory.

57. A memory control logic for accessing a re-programmable non-volatile memory comprising:

write detection logic to determine when a write operation to the memory is being performed;

address detection logic configured to indicate a detected address when a destination address associated with a write operation falls within a first range of addresses; and

write-control logic operatively coupled to the address detection logic and configured to perform a shallow-write operation of data associated with the write operation when the address detection logic indicates a detected address.

58. The memory control logic of claim 57 further comprising a first address register and a second address register, wherein the first range of addresses comprises data stored in the first and second address registers.

59. The memory control logic of claim 57 further comprising a plurality of register pairs, each register pair representative of a range of addresses in an address space of the memory, wherein the address detection logic is further configured to indicate a detected address when the destination address falls within any of the ranges of addresses represented by the plurality of register pairs.

60. The memory control logic of claim 57 wherein the re-programmable non-volatile memory is one of an EEPROM (electrically erasable programmable read-only memory) and a flash memory.

61. A memory control logic for accessing a re-programmable non-volatile memory comprising write detection logic to determine when a write operation to the memory

is being performed and write-control logic configured to perform a shallow-write operation of data associated with the write operation.

62. The memory control logic of claim 61 wherein the re-programmable non-volatile memory is one of an EEPROM (electrically erasable programmable read-only memory) and a flash memory.

63. A memory access method for accessing a re-programmable non-volatile memory comprising:

detecting a write operation to the re-programmable non-volatile memory, the write operation having an associated destination address and associated one or more data to be written to the re-programmable non-volatile memory;

determining an operating mode;

if the operating mode indicates a first mode of operation, then:

performing a partial write operation of each datum to the re-programmable non-volatile memory, if the destination address is within a first range of addresses; and

performing a full write operation of each datum to the re-programmable non-volatile memory, otherwise;

if the operating mode indicates a second mode of operation, then performing a full write operation of each datum to the re-programmable non-volatile memory;

if the operating mode indicates a third mode of operation, then performing a partial write operation of each datum to the re-programmable non-volatile memory; and

if the operating mode indicates a fourth mode of operation, then:

determining a caching location which identifies an area of memory in a dynamic memory where the data can be cached; and

writing the data in the area of memory identified by the caching location instead of writing to the re-programmable non-volatile memory.

64. The method of claim 63 wherein the re-programmable non-volatile memory is one of an EEPROM (electrically erasable programmable read-only memory) and a flash memory.